## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently amended) A computer-implemented method of mapping software components to test cases that test the software components, the method comprising:

for each component in a set of software components, generating a corresponding component node in a bipartite graph;

for each test case in a set of test cases configured to test the software components, generating a corresponding test case node in the graph; ~~and~~

for each test case:

applying said test case to test one or more software components;

collecting data reflecting application of said test case; and

updating the graph with edges coupling said corresponding test case node to one or more component nodes corresponding to software components tested by the test case; and

for each software component tested by a test case, calculating a coverage rating configured to indicate the coverage achieved from executing said test case on said software component, wherein the coverage rating is normalized to exclude dead code.

2. (Canceled)

3. (Original) The method of claim 1, further comprising:

associating said coverage rating with the edge coupling the test case node corresponding to said test case to the component node corresponding to said software component.

4. (Original) The method of claim 1, further comprising:

associating said coverage rating with the test case node corresponding to said test case.

5. (Original) The method of claim 1, further comprising:

associating said coverage rating with the component node corresponding to said software component.

6. (Original) The method of claim 1, wherein the method further comprises, for each test case:

updating said corresponding test case node with said data.

7. (Original) The method of claim 1, wherein the method further comprises, for each test case:

updating each said corresponding component node with said data.

8. (Original) The method of claim 1, wherein the method further comprises, for each test case:

marking the one or more software components.

9. (Original) The method of claim 8, wherein the method further comprises, for each test case, prior to said marking:

unmarking a software component marked during application of a previous test case.

10. (Original) The method of claim 8, wherein said marking comprises:

marking an instruction executed during application of said test case.

11. (Original) The method of claim 8, wherein said marking comprises:

marking a software component testing during application of said test case.

12. (Original) The method of claim 8, wherein said marking comprises:

marking a component module executed during application of said test case.

13. (Original) The method of claim 8, wherein said marking comprises:

marking a component function executed during application of said test case.

14. (Original) The method of claim 1, wherein said data identify one or more instructions of a software component executed during application of said test case.

15. (Original) The method of claim 1, wherein said data identify one or more modules of a software component executed during application of said test case.

16. (Original) The method of claim 1, further comprising:

using said graph to identify one or more test cases configured to test a selected software component.

17. (Currently amended) A computer readable medium storing instructions that, when executed by a computer, cause the computer to perform a method of mapping software components to test cases that test the software components, the method comprising:

for each component in a set of software components, generating a corresponding component node in a bipartite graph;

for each test case in a set of test cases configured to test the software components, generating a corresponding test case node in the graph; ~~and~~

for each test case:

applying said test case to test one or more software components;

collecting data reflecting application of said test case; and

updating the graph with edges coupling said corresponding test case node and one or more component nodes corresponding to software components tested by the test case; and

for each software component tested by a test case, calculating a coverage rating configured to indicate the coverage achieved from executing said test case on said software component, wherein the coverage rating is normalized to exclude dead code.

18. (Currently amended) A computer readable medium containing a data structure configured for mapping software components to test cases configured to test the software components, the data structure comprising:

a set of component nodes, each said component node corresponding to a software component;

a set of test case nodes, each said test case node corresponding to a test

case configured to test one or more software components; and

edges coupling each said test case node to the component nodes corresponding to the software components tested by said corresponding test case, wherein each said edge coupling a test case node and a component node has an associated rating configured to reflect the coverage achieved from executing the corresponding test case on the corresponding software component, and wherein the coverage rating is normalized to exclude dead code.

19. (Currently amended) The computer readable medium of claim 18, further comprising:

for each software component tested by a test case, a rating configured to indicate the ability of the test case to locate bugs on~~how effectively~~ the software component ~~is tested~~.

20. (Canceled)

21. (Original) The computer readable medium of claim 18, wherein each said edge is configured to identify one or more of:

a hardware configuration for executing said corresponding test case to test the corresponding software components; and

a software configuration for executing said corresponding test case to test the corresponding software components.

22. (Original) The computer readable medium of claim 18, wherein each said edge is configured to indicate an amount of time needed to execute said corresponding test case.

23. (Currently amended) A computer system for testing software, the

computer system comprising:

a set of software components;

a set of test cases configured to test the software components;

a testing tool configured to apply each test case in the set of test cases against a subset of the set of software components; ~~and~~

a graph engine configured to generate a graph comprising:

component nodes corresponding to the software component;

test case nodes corresponding to the test cases; and

edges coupling each test case node to component nodes corresponding to software components tested by the corresponding test case; and

a processor configured to generate, for each software component tested by a test case, a rating indicating the coverage achieved from executing the test case on the software component, wherein the coverage rating is normalized to exclude dead code.

24. (Canceled)

25. (Original) The computer system of claim 23, wherein each component node:

identifies one or more test cases configured to test the corresponding software component.

26. (Original) The computer system of claim 23, wherein each test case node:

identifies one or more software components tested by the corresponding test case.

27. (Original) The computer system of claim 23, wherein each edge coupling a test case node and a component node has an associated rating configured to indicate how effectively the corresponding test case covers the corresponding software component.

28. (Original) The computer system of claim 23, wherein each edge coupling a test case node and a component node:

identifies one or more of a hardware configuration and a software configuration for testing the corresponding component with the corresponding test case.

29. (Original) The computer system of claim 23, wherein each edge coupling a test case node and a component node:

identifies an amount of time needed to test the corresponding component with the corresponding test case.

30. (Original) The computer system of claim 23, wherein said testing tool is configured to:

mark a first software component when it is tested by a first test case.

31. (Original) The computer system of claim 30, wherein said testing tool is further configured to, prior to said marking:

un-mark the first software component to remove a marking made during testing of the first software component by a second test case.